REMARKS/ARGUMENTS

After the foregoing Amendment, claims 22-40 are currently pending in this application. Claims 1-21 have been cancelled without prejudice. Claims 22-40 have been added to more distinctly claim subject matter which the Applicants regard as the invention. The new claims parallel original claims 1-6 and 8-20. No new matter has been introduced into the application by this amendment.

Claim Rejections - 35 USC § 103

Claims 1, 4-7, 17-18, and 20-21 stand rejected under 35 USC § 103(a) as being unpatentable over *Kuskin* (U.S. Patent No. 6,829,683) and *Glasco* (U.S. PGPub No. 20050033924). Claims 8-12 and 14-16 stand rejected under 35 USC § 103(a) as being unpatentable over *Kuskin, Glasco,* and *McCracken* (US Patent No. 6,381,681). Those claims have been cancelled without prejudice, mooting the rejections. The new claims are deemed patentable over the cited prior art for the following reasons.

As shown in Fig. 1 and described in paragraphs [0009] *et seq.*, transfer of cache line ownership between two processors in a shared memory multi-processor computer system using a directory-based cache coherency scheme conventionally requires four transactions, as follows. A first processor sends a request for ownership of a cache line to a memory unit (10). The memory unit determines that the owner of the cache line is a second processor and sends a recall transaction to the second processor (20). The second processor returns the cache line data and ownership to the memory unit in a response transaction (30). The memory unit transfers the cache line data and ownership to the first processor (40).

The claims of the present application are directed to reducing latency in transferring cache line ownership between two processors in a shared memory multi-processor computer system using a directory-based cache coherency scheme, by reducing the number of transactions required to just three. As shown in Fig. 2 and described in paragraphs [0012] *et seq.*, a request for ownership of a cache line is sent from a first processor to a memory unit (10). The request may include a guarantee that the first processor will make the requested cache line data available in response to a subsequent request for ownership of the cache line. The memory unit receives the request and determines a second processor having ownership of the requested cache line. The memory unit sends an ownership recall to the second processor (20). In response to the ownership recall, the second processor sends the requested cache line directly to the first processor with ownership (30a, ownership is transferred). This is done regardless of whether a return request was previously sent by the second processor that has not yet been acknowledged. The first processor may then send a response to the memory unit to confirm receipt of the requested cache line (50). The second processor may also send a response to the memory unit to confirm that it sent the requested cache line to the first processor (30b). A copy of the data for the requested cache line may be sent to the memory unit by the second processor as part of its response, but this need not be done if the first processor guaranteed in its request for ownership of the cache line that it will make the cache line data available in response to a subsequent request for ownership of the cache line.

In contrast, *Kuskin* teaches transferring ownership of data in a distributed shared memory system, wherein a first processor has modified a cache line it owns and wishes to update a memory directory associated with the home directory of the cache line. The first processor generates a return request for routing by a processor interface to the memory directory. A

second processor wishing to obtain ownership of the cache line sends a read request to the memory directory. In response, the memory directory generates an intervention request towards the processor interface. By the time the processor interface receives the intervention request, it has already forwarded the return request to the memory directory. In this narrow circumstance, when the processor interface receives the intervention request for the second processor, rather than waiting for an acknowledgement from the memory directory that its return request has been processed, the processor interface instead sends an intervention response to the second processor that includes the modified cache line. (See, e.g., *Kuskin* Abstract, column 1 lines 38-50.) In general, however, the processor interface does not send the modified cache line directly to the second processor in response to an intervention request. Rather, the transfer of cache line ownership would presumably require 4 transactions as described above.

The Examiner cites *Kuskin* col. 2 line 63 through col. 3 line 2 for sending the requested cache line with ownership from the second processor to the first processor in response to the recall transaction. However, *Kuskin* in the cited location teaches only that the data may be "retrieved" from the second processor. *Kuskin* then goes on in col. 3 lines 2-6 to teach that the memory directory responds to incoming message from anywhere in the system, and updates the state of a particular cache line and generates messages in response to the incoming messages, presumably, as to transferring cache line ownership, requiring 4 transactions as described above. Nothing in these passages discloses or suggests sending the requested cache line with ownership directly from the second processor to the first processor in response to a recall transaction, as do the independent claims of the present application. In addition, *Kuskin* is further easily distinguished from the independent claims of the present application, wherein the requested

cache line with ownership is sent directly from the second processor to the first processor regardless of whether an unacknowledged return request was sent by the second processor.

As to *Glasco*, *Glasco* teaches an architecture wherein multiple processors are used in a system sharing the same memory space using point-to-point links between processors. However, as the number of processors in such an architecture is increased, the number of links used to connect the processors increases exponentially. In order to reduce the number of links, the processors are organized into clusters, and the clusters themselves are interconnected using a point-to-point architecture. Each cluster includes a cache coherence controller, used to handle communications between clusters. (*Glasco* paragraph [0030].) Each cache coherence controller is associated with a so-called remote data cache that receives data and state information for memory lines held in remote clusters (see, e.g., *Glasco* abstract), and represents non-local nodes in local transactions. The remote data caches are basically an additional layer of caching (*Glasco* paragraph [0037]).

In contrast to all independent claims of the present application, *Glasco* does not appear to pertain to a directory-based cache coherency scheme, such as one that tracks which processors own which cache lines. For example, *Glasco* describes transactions for a cache request from a processor in a system having a single cluster without using a cache coherence controller. A first processor sends an access request such as a read memory line request to a memory controller, which blocks or queues subsequent requests to the same memory line and sends probes to all local cache memories, such as those associated with other processors, to determine cache states. The local cache memories all then send probe responses to the processor, one of which presumably contains the requested cache line. The requesting processor then sends a done response to the memory controller, which unlocks the memory line for subsequent requests

(*Glasco* paragraphs [0049]-[0050]). In a directory-based cache coherency scheme, the memory controller typically would contact only the processor having ownership of the requested cache line, and would not contact all local processors.

From this point, *Glasco* might have suggested, but instead failed to suggest, that a directory-based cache coherency scheme could be used to identify and send a transaction to the processor having ownership of the requested cache line, which could then send the cache line with ownership directly to the requesting processor. Instead, and in dramatic contrast to the claims of the present application, *Glasco* introduces a cache coherence controller (*Glasco* paragraph [0051] et seq.), which "allows the creation of a multiprocessor, multicluster coherent domain without affecting the functionality of local nodes such as processor and memory controllers in each cluster." The cache coherence controller is used to make local processors believe that all non-local nodes are a single local node embodied in the cache coherence controller, which represents the non-local nodes.

To achieve this result, *Glasco* provides so-called "remote data caches" associated with the various cache coherence controllers, which receive and store cache lines and state information for the processors in their associated clusters. The remote data caches are used to facilitate transferring ownership of cache lines between processors of different clusters. A cache line request is received at a cache coherence controller from a processor in a cluster of processors associated with the cache coherence controller. If information for responding to the cache line access request is available in a remote data cache associated with the cache coherence controller, a response is provided by the remote data cache to the requesting processor. In contrast to the present application, response information is obtained from the remote data cache, which provides it to the requesting processor. The response information is not received directly from a second

processor that owns the cache line, as in the claims of the present application. (See, e.g, *Glasco* paragraphs [0009]-[0010].)

In contrast to *Glasco*, the claims of the present invention effectively reduce the amount of caching that is required to track and transfer ownership of cache lines. *Glasco* essentially does the opposite, by organizing a plurality of processors into clusters, and introducing an entire additional layer of caching associated with the clusters. The Examiner cites *Glasco* for various features of the claims of the present application, such as the requesting processor confirming receipt of the requested cache line, and sending a response transaction from the second processor to the memory unit to confirm that the second processor has sent the requested cache line to the first processor. The Examiner then combines those features with *Kuskin* in an attempt to arrive at the claims of the present application.

However, Applicants respectfully point out that *Glasco* is not directed to the interactions of processors requesting and directly providing cache line ownership, but rather with the interactions of cache coherence controllers, which are associated with remote data caches, which are themselves associated with clusters of processors, one of which comprises a processor requesting a cache line, and another of which comprises a processor having the requested cache line, all without the use of a directory-based cache coherency scheme. It is hardly clear how one would combine such an arrangement with a directory-based cache coherency scheme to arrive at anything like the present claims. Neither *Kuskin* nor *Glasco* teach or suggest combining their features, and applicants respectfully submit that one of ordinary skill in the art would not have been motivated to combine the references relied upon by the Examiner in the manner suggested by the Examiner, except in hindsight in view of the present application.

Furthermore, as to claim 26 (similar to original claim 5), none of the references relied upon by the Examiner disclose or suggest, either alone or in any possible combination thereof, a response transaction from the second processor to the memory unit that includes a copy of the data for the requested cache line only when the request for ownership of the cache line from the first processor does not include a guarantee that the first processor will make the requested cache line data available in response to a subsequent request for ownership of the cache line. Even though the Examiner asserts that *Kuskin* does disclose this feature, the Applicants respectfully traverse this assertion. At the cited location (*Kuskin* col. 22 lines 1-4), *Kuskin* is discussing the handling of an intervention that arrives when a write request message is outstanding, and not whether a response transaction from a second processor should include a copy of the data for the requested cache, depending on whether or not a request for cache line ownership from a first processor included a guarantee from the first processor that it will make the requested cache line available in response to a subsequent request for ownership. *Kuskin* does not disclose anything having to do with this feature, either at the location cited by the Examiner or elsewhere. Similarly, none of the references relied upon by the Examiner disclose or suggest, either alone or in any possible combination thereof, a response transaction from the second processor to the memory unit that includes a copy of the data for the requested cache line only when the request for ownership of the cache line from the first processor does not include a guarantee that the first processor will make the requested cache line data available in response to a subsequent request for ownership of the cache line, as claimed.

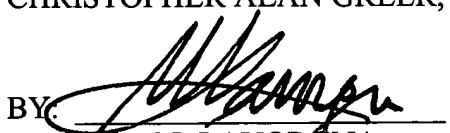
Conclusion

In view of the foregoing amendment and remarks, Applicant respectfully submits that the present application, including claims 22-40, is in condition for allowance and a notice to that effect is respectfully requested.

If the Examiner believes that any additional minor formal matters need to be addressed in order to place this application in condition for allowance, or that a telephone interview will help to materially advance the prosecution of this application, the Examiner is invited to contact the undersigned by telephone at the Examiner's convenience.

Respectfully submitted,

CHRISTOPHER ALAN GREER, *ET AL.*



BY: _____
GREGORY J. LAVORGNA
Registration No. 30,469
DRINKER BIDDLE & REATH LLP
One Logan Square
18th and Cherry Streets
Philadelphia, PA 19103-6996
Tel: 215-988-3309
Fax: 215-988-2757
Attorney for Applicant